

Documentación ApiVass

URL_BASE = <http://localhost:3002>

Auth

Endpoint: (URL_BASE /auth/login) / Method: **POST**

Descripción: Inicia sesión en la app, la cual devuelve un token de acceso.

Parámetros:

- Email (**requerido**)
- Password (**requerido**)

Ej. JSON:

```
{
  email: "test123@gmail.com",
  password: "TestPassword123"
}
```

Respuesta:

```
{
  message: "Usuario logueado",

  token:
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQsw5c",

  user: {
    _id: "64774ef7f70d53f1924d0dd7",
    name: "test";
    email: "test123@gmail.com";
    password: "$2b$10$.0cQ9pHUJyoNPaNbZBW0c.aEEkIuLrDcqA094GiFAXu5CRVipR91G";
    age: 20;
    city: "Cali";
  }
}
```

Token: JWT Token de acceso que será usado posteriormente en los demás endpoints para validar la autenticación del usuario.

Users

Obtener todos los usuarios

Endpoint: (URL_BASE /users) / Method: **GET**

Descripción: Devuelve todos los usuarios registrados en el sistema.

Respuesta:

```
[
  {
    _id: "648515b2d447d6de72719261",
    name: "Test user",
    email: "testuser@gmail.com",
    password:
"$2b$10$.0cQ9pHUJyoNPaNbZBW0c.aEEkIuLrDcqA094GiFAXu5CRHipR91G",
    city: "Bogotá",
    __v: 0,
  },
  {
    _id: "648515b2d447d6de72719261",
    name: "Test user",
    email: "testuser@gmail.com",
    password:
"$2b$10$.0cQ9pHUJyoNPaNbZBW0c.aEEkIuLrDcqA094GiFAXu5CRHipR91G",
    city: "Bogotá",
    __v: 0,
  },
  . . .
]
```

Obtener la información de un usuario por su id

id = 6483bb205245e3862d83cf61 (requerido)

Endpoint: (URL_BASE /users/id) / Method: **GET**

Descripción: Devuelve la información de un usuario registrado en el sistema a través de su identificador único (id)

Respuesta:

```
{
  _id: "648515b2d447d6de72719261",
  name: "Test user",
  email: "testuser@gmail.com",
  password: "$2b$10$.0cQ9pHUJyoNPaNbZBW0c.aEEkIuLrDcqA094GiFAXu5CRHipR91G",
  city: "Bogotá",
  __v: 0,
},
```

Crear un nuevo usuario

Endpoint: (URL_BASE /users/new) / Method: **POST**

Descripción: Crea un nuevo usuario en la base de datos del sistema, con la contraseña encriptada

Parámetros:

- Email (**requerido**)
- Password (**requerido**)
- Name (**requerido**)
- Age (opcional)
- City (opcional)

```
JSON
{
  name: "Test user",
  email: "testuser@gmail.com",
  password: "TestUser123",
  city: "Bogotá",
  age: 20
}
```

Respuesta:

```
{
  message: "User created successfully",
  {
    _id: "648515b2d447d6de72719261",
    name: "Test user",
    email: "testuser@gmail.com",
    password: "$2b$10$.0cQ9pHUJyoNPaNzBWOC.aEEkIuLrDcqA094GiFAXu5CRHipR91G",
    city: "Bogotá",
    age: 20,
    __v: 0,
  }
}
```

Actualizar un usuario existente por su id

Endpoint: (URL_BASE /users/update/id) / Method: **PUT**

Descripción: Actualiza un usuario existente en el sistema a través de su identificador único (id)

Parámetros:

id = 6483bb205245e3862d83cf61 (**requerido**)

- Email (opcional)
- Password (opcional)
- Name (opcional)
- Age (opcional)
- City (opcional)

```
JSON
{
  city: "Cali",
  age: 21
}
```

Respuesta:

```
{
  message: "User updated successfully",
  {
    _id: "648515b2d447d6de72719261",
    name: "Test user",
    email: "testuser@gmail.com",
    password: "$2b$10$.0cQ9pHUJyoNPaNbZBWOC.aEEkIuLrDcqA094GiFAXu5CRHipR91G",
    city: "Cali",
    age: 21,
    __v: 0,
  }
}
```

Eliminar un usuario existente por su id

id = 6483bb205245e3862d83cf61 (**requerido**)

Endpoint: (URL_BASE /users/delete/id) / Method: **DELETE**

Descripción: Elimina un usuario existente en el sistema a través de su identificador único (id)

Respuesta:

```
{
  message: "User deleted successfully",
  {
    _id: "648515b2d447d6de72719261",
    name: "Test user",
    email: "testuser@gmail.com",
    password: "$2b$10$.0cQ9pHUJyoNPaNzBW0c.aEEkIuLrDcqA094GiFAXu5CRHipR91G",
    city: "Cali",
    age: 21,
    __v: 0,
  }
}
```